Regularization-Based Efficient Continual Learning in Deep State-Space Models

Yuanhang Zhang†, Zhidi Lin†, Yiyong Sun†, Feng Yin†(✉), and Carsten Fritsche‡
† School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
‡ IAV GmbH, Weimarer Str. 10, 80807 Muenchen, Germany

***Abstract*—Deep state-space models (DSSMs) have gained popularity in recent years due to their potent modeling capacity for dynamic systems. However, existing DSSM works are limited to single-task modeling, which requires retraining with historical task data upon revisiting a forepassed task. To address this limitation, we propose continual learning DSSMs (CLDSSMs), which are capable of adapting to evolving tasks without catastrophic forgetting. Our proposed CLDSSMs integrate mainstream regularization-based continual learning (CL) methods, ensuring efficient updates with constant computational and memory costs for modeling multiple dynamic systems. We also conduct a comprehensive cost analysis of each CL method applied to the respective CLDSSMs, and demonstrate the efficacy of CLDSSMs through experiments on real-world datasets. The results corroborate that while various competing CL methods exhibit different merits, the proposed CLDSSMs consistently outperform traditional DSSMs in terms of effectively addressing catastrophic forgetting, enabling swift and accurate parameter transfer to new tasks.**

***Index Terms*—Continual learning, state-space model, deep learning, regularization, efficient learning.**

I. Introduction

State-space models (SSMs) provide a fundamental framework for system identification and state inference in dynamic systems [1]. With their excellent model interpretability, diverse SSMs have found successful applications across various domains in recent decades, including robotic control, healthcare, climate change tracking and indoor positioning [2]–[10]. Nevertheless, classic SSMs generally require prior knowledge of the underlying system dynamics, which are challenging to determine in advance. Consequently, there is a need to learn the dynamics from observed noisy measurements, giving rise to the development of data-driven SSMs [6]–[11].

One of the popular data-driven SSMs is deep state-space models (DSSMs), which leverage deep neural networks as their central modeling component, augmenting the learning and inference capabilities of classic SSMs and reducing the reliance on the modeling prior knowledge [11]–[15]. Due to these advantageous properties, significant strides have been achieved in DSSMs, encompassing applications such as time series prediction [11]–[13], nonlinear system identification [14]–[16], and healthcare [10], [11].

However, despite the notable performance of DSSMs in isolated tasks, their practical usage in real-world scenarios is likely to be hindered by limitations in memory and computing resources. This is because practical applications often involve successive arriving data, which further exacerbates the resource constraints [17]. These constraints necessitate DSSMs to exhibit continual learning (CL) capabilities. For instance, in domains like autonomous driving systems [18], where DSSMs are expected to navigate in dynamic environments, and continuously track without frequent model retraining or extensive storage of historical task data. Consequently, the development of DSSMs capable of continual learning becomes a critical step to quickly adapt to and learn from multiple tasks [19]. It is noted that, unlike multi-task learning, which jointly addresses multiple offline tasks [20], and transfer learning, which involves transferring knowledge from one task to another [21], continual learning emphasizes a model's capacity to learn continuously from sequential data streams. For further insights, one can refer to, e.g., [22].

Recent research on continual learning predominantly concentrates on supervised learning and can be broadly classified into three main approaches: replay [23]–[25], regularization [26]–[32], and parameter isolation [33]. More specifically, replay methods involve replaying a subset of previous data, regularization methods employ regularization terms to consolidate historical knowledge, and parameter isolation methods enable the learning model to develop new branches for future tasks while preserving parameters for previous tasks.

With our aim to reduce the storage and computational overhead of DSSMs in continual learning, in this paper, we focus on the regularization-based approaches to eliminate the need for storing raw inputs akin to the replay methods. In addition, the regularization-based approaches can preserve the conciseness of the deep neural network in DSSMs, rendering a memory-efficient learning framework compared to the parameter isolation approaches. Our main contributions are summarized as follows:

- We incorporate regularization-based continual learning methods into DSSMs, resulting in continual learning DSSMs (CLDSSMs). The proposed CLDSSMs demonstrate the capability to continually learn multiple dynamic systems without encountering catastrophic forgetting issues, rendering them adaptable to a wider range of system modeling applications. To the best of our knowledge, this paper marks the first exploration of continual learning in the context of DSSMs.
- We demonstrate that the CLDSSM, enhanced with var-

This work was supported by the Shenzhen Science and Technology Program under Grant No. JCYJ20220530143806016, and in part by NSFC under Grant No. 62271433. Feng Yin (*yinfeng@cuhk.edu.cn*) is the Corresponding Author.

ious prevalent regularization-based continual learning methods, maintains a constant memory cost even with a continually expanding volume of data. Our approach also achieves superior results without requiring training with historical data, thus addressing the limitation of the standard DSSM. Furthermore, the continual learning methods employed in CLDSSMs acquire knowledge from historical tasks and thus can help expedite the training when encountering new related tasks, enabling the model to attain satisfactory results in earlier training phases.

- We evaluate the performance of the proposed CLDSSMs on real-world datasets, showcasing their efficacy in overcoming catastrophic forgetting while achieving savings in both computational and memory costs, and enhancing model training. These results highlight the versatility of the CLDSSMs and their applicability to various real-world dynamic system modeling applications.

The subsequent sections of this paper are structured as follows. Some preliminaries and background about DSSMs are presented in Section II. Section III outlines the pipeline of the proposed CLDSSMs. Numerical results are presented in Section IV, and Section V concludes this paper.

II. Preliminaries

This section begins with a brief introduction to DSSMs in Section II-A. Subsequently, Section II-B reviews a learning method based on the autodifferentiable ensemble Kalman filter (EnKF) [34].

A. Deep State-Space Models

As depicted in Fig. 1, we consider an SSM that characterizes the probabilistic relationship between the latent state $\mathbf{z}_t \in \mathbb{R}^{d_z}$ and the observation $\mathbf{x}_t \in \mathbb{R}^{d_x}$, as expressed by the following equations:

$$\mathbf{z}_t = F_\alpha(\mathbf{z}_{t-1}) + \xi_t, \qquad \xi_t \sim \mathcal{N}(0, Q_\beta), \tag{1}$$

$$\mathbf{x}_t = H\mathbf{z}_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, R), \tag{2}$$

where $F_\alpha(\cdot)$ is the transition function that maps the latent state $\mathbf{z}_{t-1}$ to the future state $\mathbf{z}_t$, with $1 \leq t \leq T$. The emission function (see Eq. (2)) that maps latent states to observations is assumed to be linear and known, with the coefficient matrix $H \in \mathbb{R}^{d_x \times d_z}$. The noises ξ_t and η_t are additive and independent Gaussian random variables. The model parameters, denoted by $\theta \triangleq \{\alpha, \beta\}$, are both unknown and time-invariant. When the transition function is modeled using deep neural networks, the resulting model is referred to as a deep SSM (DSSM) [14]. Within DSSM, α represents the parameter associated with the transition neural network, while β is the covariance matrix of the Gaussian noise ξ_t.

In DSSMs, one of the most challenging tasks is to simultaneously learn the model parameter θ and infer the latent state of interest $\mathbf{z}_t$. This typically involves dealing with the model marginal likelihood [35], [36], which can be expressed mathematically as:

$$p_\theta(\vec{\mathbf{x}}) = \int p_\theta(\vec{\mathbf{x}}, \vec{\mathbf{z}})\, \mathrm{d}\vec{\mathbf{z}} = \int p(\vec{\mathbf{x}}|\vec{\mathbf{z}}) p_\theta(\vec{\mathbf{z}}) \mathrm{d}\vec{\mathbf{z}}, \tag{3}$$

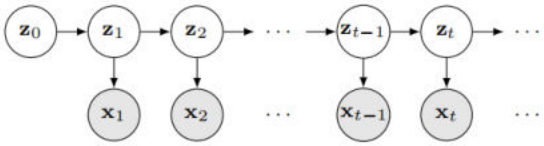


Fig. 1: Graphical representation of an SSM.

where $\vec{\mathbf{x}} \triangleq \{\mathbf{x}_t\}_{t=1}^T$ and $\vec{\mathbf{z}} \triangleq \{\mathbf{z}_t\}_{t=1}^T$ represent the sequences of observations and latent states of length T, respectively. The term $p_\theta(\vec{\mathbf{z}})$ is the prior distribution of the latent states, and $p(\vec{\mathbf{x}}|\vec{\mathbf{z}})$ represents the model emission or likelihood function. Since the integral in Eq. (3) is generally intractable, further approximation is required.

B. Autodifferentiable Ensemble Kalman Filters

Numerous approximation techniques exist for addressing the intractable model evidence in Eq. (3), including variational inference, MCMC sampling, Laplace, among others [11]–[16], [35], [36]. This paper predominantly employs a very recent approximation approach based on the ensemble Kalman filter (EnKF) [37], [38], namely the autodifferentiable EnKF [34]. The autodifferentiable EnKF leverages the well-established EnKF for state inference while exploiting the autodifferentiation feature to simultaneously learn the parameter θ. Specifically, to derive the marginal likelihood $p(\vec{\mathbf{x}})$, the EnKF propagates N equally weighted particles, denoted as $\mathbf{z}_{t-1}^{1:N}$, from the filtering distribution, $p_\theta(\mathbf{z}_{t-1}|\mathbf{x}_{1:t-1})$, see Eq. (6), using the transition function to approximate the forecast distribution, $p_\theta(\mathbf{z}_t|\mathbf{x}_{1:t-1})$:

$$p_\theta(\mathbf{z}_t|\mathbf{x}_{1:t-1}) = \int p_\theta(\mathbf{z}_t|\mathbf{z}_{t-1}) p_\theta(\mathbf{z}_{t-1}|\mathbf{x}_{1:t-1})\, \mathrm{d}\mathbf{z}_{t-1}, \tag{4a}$$

$$\approx \mathcal{N}(\widehat{m}_t, \widehat{C}_t), \quad \cdots\cdots \text{ (forecasting step)} \tag{4b}$$

where

$$\widehat{m}_t = \frac{1}{N}\sum_{n=1}^N \widehat{\mathbf{z}}_t^n, \tag{5a}$$

$$\widehat{C}_t = \frac{1}{N-1}\sum_{n=1}^N (\widehat{\mathbf{z}}_t^n - \widehat{m}_t)(\widehat{\mathbf{z}}_t^n - \widehat{m}_t)^\top, \tag{5b}$$

and $\widehat{\mathbf{z}}_t^n = F_\alpha(\mathbf{z}_{t-1}^n) + \xi_t^n$, $n = 1, 2, \ldots, N$, denotes the forecast ensemble.

Then, due to the linearity and the Gaussian nature of the emission model, we can recursively obtain the filtering distribution at time step t, $p_\theta(\mathbf{z}_t \mid \mathbf{x}_{1:t})$:

$$p_\theta(\mathbf{z}_t|\mathbf{x}_{1:t}) = \mathcal{N}(m_t, C_t), \quad \cdots\cdots \text{ (filtering step)} \tag{6a}$$

$$m_t = \widehat{m}_t + \widehat{K}_t(\mathbf{x}_t - H\widehat{m}_t), \tag{6b}$$

$$C_t = \widehat{C}_t - \widehat{C}_t\widehat{K}_t^\top\widehat{C}_t^\top, \tag{6c}$$

where $\widehat{K}_t \triangleq \widehat{C}_t H^\top (H\widehat{C}_t H^\top + R)^{-1}$ represents the Kalman gain. With these steps, the logarithm of the marginal likelihood thus can be approximated as

$$\mathcal{L}_{\mathrm{EnKF}}(\theta) \triangleq \log p_\theta(\vec{\mathbf{x}}) = \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t|\mathbf{x}_{1:t-1}) \tag{7a}$$

$$= \sum_{t=1}^{T} \log \int p(\mathbf{x}_t|\mathbf{z}_t) p_\theta(\mathbf{z}_t|\mathbf{x}_{1:t-1})\, \mathrm{d}\mathbf{z}_t \tag{7b}$$

$$= \sum_{t=1}^{T} \log \mathcal{N}\left(H\widehat{m}_t, H\widehat{C}_t H^\top + R\right). \tag{7c}$$

Leveraging the reparameterization trick [39] in forecast and filtering steps, the autodifferentiable EnKF constructs the map $\theta \mapsto \mathcal{L}_{\mathrm{EnKF}}(\theta)$ [34]. Consequently, we can optimize the θ via gradient descent-based methods. Yet, all existing learning methods, including the autodifferentiable EnKF, exclusively address single-task scenarios and lack suitability for managing multiple tasks. Our focus will shift to exploring continual learning in DSSMs in the next section.

III. Continual Learning in DSSMs

This section elaborates on the main ingredients of our proposed method. As described in Fig. 2, we consider $J \in \mathbb{N}$ interrelated dynamic modeling tasks, each observed sequentially. The associated datasets cannot be stored due to limited storage memory. Specifically, we denote the observable data of size T_j for the j-th dynamic modeling task as $\vec{\mathbf{x}}_j = \{\mathbf{x}_{j,t}\}_{t=1}^{T_j}$, $j = 1, \ldots, J$. Given the observed data, the objective for the DSSM is to continually learn the underlying dynamic systems without experiencing catastrophic forgetting. This ensures that the learned DSSM can be utilized to make predictions across various dynamic modeling tasks. Additionally, we aim to maintain a constant level of memory consumption for model training, irrespective of the number of tasks.

As mentioned in Section I, this paper focuses on regularization-based CL methods, of which the general idea involves introducing a weighted regularization term to the original loss to penalize deviations in model parameters when new tasks arise. Various established regularization-based CL methods will be integrated into the DSSM [26]–[30], except for variational continual learning (VCL) [32]. This exclusion is motivated by research findings indicating that learning and inference in the Bayesian neural network in VCL can be computationally expensive [35], [40]. In the following subsections, we elaborate on how the regularization-based CL methods empower DSSMs to continually learn the underlying system dynamics.

A. Learning the First Task

We begin with the introduction to the learning of the first task, which involves classic learning and inference in DSSMs, as presented in Section II-B. Specifically, we first evaluate the log-likelihood, see Eq. (7) through the propagation of the emission function using the approximated forecast distribution

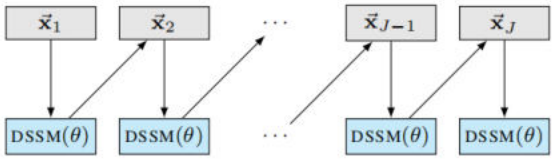


Fig. 2: Illustration of continual learning in DSSMs

obtained from EnKF. Then, a recognition network is introduced to infer the initial latent state from input observations [8], [11]. More concretely, we approximate the posterior distribution of the initial latent state using the introduced parametric distribution, $q_\phi(\mathbf{z}_{1,0}|\vec{\mathbf{x}}_1)$, where $\mathbf{z}_{1,0}$ denotes the initial latent state of the first task. The distribution $q_\phi(\mathbf{z}_{1,0}|\vec{\mathbf{x}}_1)$ is assumed to be Gaussian, with mean (μ_ϕ) and variance (Σ_ϕ) functions modeled by a recurrent neural network (RNN)-based recognition network with parameters ϕ. Therefore, the loss function aimed at minimizing for the first task is expressed as follows:

$$\mathcal{L}(\vec{\mathbf{x}}_1; \theta, \phi) = \sum_{t=1}^{T} \log \mathcal{N}\left(H\widehat{m}_t, H\widehat{C}_t H^\top + R\right) \tag{8a}$$

$$- \mathrm{KL}\left[q_\phi(\mathbf{z}_{1,0}|\vec{\mathbf{x}}_1) \| p(\mathbf{z}_{1,0})\right], \tag{8b}$$

where Eq. (8a) represents the model log-likelihood function, optimizing the observation reconstruction capability, and Eq. (8b) corresponds to the Kullback-Leibler (KL) divergence between the posterior distributions and the prior distributions, which serves as a regularizer, ensuring that the posterior distributions do not deviate significantly from the prior distributions, facilitating the learning of the system dynamics [13]. The prior $p(\mathbf{z}_{1,0})$ is assumed to adhere to a known Gaussian distribution. Due to the utilization of this reparameterization trick [34], we can optimize the model parameter θ via a modern optimizer, such as Adam [41].

B. Continual Learning for Subsequent Tasks

This subsection details various regularization-based continual learning methods for the DSSM to continually learn the model parameters $\theta = (\alpha, \beta)$ in the subsequent tasks [26]–[30]. Note that the inference network parameters ϕ are task-specific and play a relatively marginal role in DSSM [34]. Therefore, we do not discuss their continual learning but rather train them independently in each task.

a) Elastic Weight Consolidation (EWC): The fundamental concept of EWC revolves around controlling the deviation of parameters that are deemed crucial for previous tasks [26], [27]. Through the addition of an extra regularization term, the overall loss function for EWC is expressed as:

$$\mathcal{L}(\vec{\mathbf{x}}_j; \theta, \phi) = \mathcal{L}(\vec{\mathbf{x}}_j; \theta, \phi) + \sum_{k=1}^{j-1} \left(\frac{\lambda}{2} \sum_i M_{k,i} \left(\theta_i - \theta^*_{k,i}\right)^2 \right), \tag{9}$$

where λ is a weighting hyperparameter that represents the importance of previous tasks, and M_k is the diagonal Fisher information matrix for the k-th task evaluated at θ^*_k, which

quantifies the importance of each model parameter with respect to the corresponding task [42]. The parameter learned from the k-th task is denoted as θ_k^*, and the subscript i denotes the i-th parameter in the model.

However, the number of quadratic regularization terms grows with the number of tasks in the vanilla EWC [26], resulting in escalating memory and computation requirements. Online-EWC addresses this limitation by defining the following objective function:

$$\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)=\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)+\frac{\lambda}{2}\sum_i\left(\tilde{M}_{j-1,i}\left(\theta_i-\theta_{j-1,i}^*\right)^2\right). \quad (10)$$

Unlike the vanilla EWC, see Eq. (9), the quadratic regularization term in Online-EWC, see Eq. (10), is a single moving sum on each related parameter, i.e., $\tilde{M}_j=\gamma\tilde{M}_{j-1}+M_j$, where $\tilde{M}_1=M_1$ and γ ($\gamma\leq 1$) is a hyperparameter governing the contribution of each previous task. Notably, after training the $(j-1)$-th task ($j>1$), the network parameters are already optimal for all preceding tasks. Consequently, there should be only one regularization term anchored at the parameters learned from the latest task (which is denoted as θ_{j-1}^* in Eq. (10)). Due to the advantageous on-memory computational characteristics of Online-EWC, the subsequent sections of this paper solely focus on Online-EWC, which will be referred to as EWC throughout the remainder of this paper for simplicity.

b) Memory Aware Synapses (MAS): Similar to EWC, the regularization term in MAS is derived from the gradient of the learned function, and the importance of model parameters can be updated in an online fashion [28]. Specifically, the loss function of MAS is expressed as:

$$\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)=\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)+\lambda\sum_i\left(\Omega_{j-1,i}\left(\theta_i-\theta_{j-1,i}^*\right)^2\right), \quad (11)$$

where λ is the hyperparameter with a similar role as in EWC; $\Omega_j=\frac{1}{T_j}\sum_{t=1}^{T_j}\parallel g(\mathbf{z}_{j,t})\parallel$ represents the parameter importance matrix and controls the deviation of model parameters, where $g(\mathbf{z}_{j,t})$ is the gradient of the transition evaluated at each state $\mathbf{z}_{j,t}$ with respect to the parameter θ.

c) Synaptic Intelligence (SI): In comparison to previous methods, the update process for the importance of SI is more complicated [29]. The loss function of SI is given by:

$$\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)=\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)+\lambda\sum_i\left(\Lambda_{j-1,i}\left(\theta_i-\theta_{j-1,i}^*\right)^2\right), \quad (12)$$

where λ ($\lambda\leq 1$) is a hyperparameter that balances the influence between tasks, and Λ_{j-1} denotes the importance matrix for the previous $j-1$ tasks and can be represented as:

$$\Lambda_{j-1,i}=\sum_{k=1}^{j-1}\frac{\omega_{k,i}}{(\Delta_{k,i})^2+\epsilon}, \quad (13)$$

where $\Delta_k=\theta_k[I]-\theta_k[0]$ represents the deviation of parameters after total I iterations of training on the k-th task, and ω_k indicates the importance of each parameter to the loss and is obtained by the product of two gradients, i.e., $\omega_k=\frac{\partial\mathcal{L}(\vec{\mathbf{x}}_k)}{\partial\theta_k}\frac{\partial\theta_k}{\partial t}$. Note that here ω_k is updated in every iteration, while Δ_k is only updated after I iterations. The parameter ϵ serves as a damping parameter in the case where $\Delta_k\to 0$, and it is set to $\epsilon=0.01$ by default [29].

d) Learning without Forgetting (LwF): The regularization utilized in LwF takes a different approach from the aforementioned methods, which is often termed functional regularization rather than parameter regularization [30]. Unlike the original LwF, which takes a distillation loss as the regularization term for classification problems, we use the mean squared error (MSE) loss instead given the observations are continuous. Specifically, the loss function with LwF is given by:

$$\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)=\mathcal{L}(\vec{\mathbf{x}}_j;\theta,\phi)+\frac{\lambda}{j-1}\sum_{k=1}^{j-1}\mathcal{L}_{\text{MSE}}\left(\hat{\mathbf{x}}_k(\theta),\hat{\mathbf{x}}_k(\theta_{j-1}^*)\right), \quad (14)$$

where $\hat{\mathbf{x}}_k(\theta_{j-1}^*)$ denotes the forecast observation sequence for the k-th task, $k=1,2,\dots j-1$, using the DSSM with parameter θ_{j-1}^*, while $\hat{\mathbf{x}}_k(\theta)$ is the forecast observation generated with the updated model parameters, and the $\mathcal{L}_{\text{MSE}}(\cdot,\cdot)$ represents the standard MSE loss [35]. The length of the forecast observation sequence is the same as the test dataset of each task.

C. Computational and Memory Cost Analysis

The overall computational and memory cost of the CLDSSM arises from the autodifferentiable EnKF and the additional regularization. The cost associated with the autodifferentiable EnKF is task-independent, with a memory cost of $\mathcal{O}(d_zN)$ and a computational cost of $\mathcal{O}(d_zd_xN)$ for the Kalman Gain evaluation [38]. The complexity of ensemble forward propagation depends on the structure of the deep neural network-based transition function within the DSSM.

Regarding the regularization part, the memory costs for EWC, MAS, and SI are consistent with $\mathcal{O}({d_z}^2)$, while LwF incurs a memory cost of $\mathcal{O}(d_zT)$, where T denotes the size of the forecast data. In terms of computational costs, both EWC and MAS scale as $\mathcal{O}({d_z}^2)$, while SI scales as $\mathcal{O}({d_z}^3)$, and the cost of LwF is dominated by the computation of MSE part, which scales as $\mathcal{O}(d_zT)$.

To be more specific, under general conditions where $d_z\ll T$, the memory costs of different methods can be ranked as EWC=MAS=SI<LwF, while the computational costs are EWC=MAS<SI<LwF. The parameter regularization methods, including EWC, MAS, and SI, control the deviation of parameters through different gradients and are expected to be effective in retaining data features under multi-task training. Meanwhile, the functional regularization method, LwF, constrains parameter training via the MSE function, likely resulting in better performance in numerical evaluations during experiments. Moreover, although the LwF method incurs significantly higher storage usage, its structure and computation are more concise and succinct compared to the other methods [30].

It is also noteworthy that the computational and memory costs of all regularization methods are independent of the

number of tasks and the size of training data. Consequently, the corresponding CLDSSM is highly efficient and does not exhibit the issue of computational and memory escalation as in conventional DSSMs.

IV. Experiments

We assess the effectiveness of the proposed CLDSSMs across two real-world datasets, demonstrating their ability to overcome catastrophic forgetting issues while keeping a constant memory and computational cost. The main evaluation metric employed for this assessment is the MSE of the model observation predictions, i.e.,

$$\text{MSE} \triangleq \mathbb{E}\left[\|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2\right], \tag{15}$$

where $\mathbf{x}$ represents the ground truth, and $\widehat{\mathbf{x}}$ is the predictive output. For each individual dataset, the training hyperparameter of each method is consistent across all tasks, indicating that each task is assigned equal importance. In our experiment, we set $\lambda_{\text{EWC}} = 1000$ for EWC, $\lambda_{\text{MAS}} = 800$ for MAS, $\lambda_{\text{SI}} = 1$ for SI and $\lambda_{\text{LwF}} = 1$ for LwF. The optimizer used is Adam [41], with a learning rate of 0.005.

A. Power Consumption Dataset

We first evaluate CLDSSMs using a real-world power consumption dataset obtained from Tetouan city [43]. The dataset spans the year 2017 with a 10-minute time interval and comprises 52,416 samples with five-dimensional control inputs and three-dimensional observations. The observations include recorded power consumption in three distribution networks in Tetouan, influenced by diverse inputs such as temperature, humidity, wind speed, diffuse flow, and general diffuse flow.

To assess the proficiency of CLDSSMs in continually learning multiple tasks, we divide the dataset into four segments chronologically, corresponding to four learning sequential tasks. Subsequently, we randomly select 1800 successive samples from each segment to form 32 training target sequences with a sequence length of $T = 50$. The remaining 200 samples in each segment are reserved as the test data for predictive analysis.

We compare the various CLDSSMs with the baseline DSSM, and Table I reports the prediction MSE, where the averaged MSE is evaluated using all the test data from the trained tasks. Thus, the averaged MSE can indicate the effectiveness of our approach in overcoming catastrophic forgetting. Our results highlight the substantial performance improvement of CLDSSMs over DSSM, especially when dealing with multiple tasks. Notably, the MSE values sharply increase during the 3rd and 4th tasks when training with the baseline DSSM, indicating significant challenges in transferring parameters from old to new tasks. In contrast, all of our proposed CLDSSMs consistently maintain lower MSE values, showcasing their ability to mitigate catastrophic forgetting. Furthermore, the smaller standard deviations in CLDSSMs results indicate enhanced stability compared to the baseline.

The more detailed results of various models are depicted in Fig. 3, where the predictions are made for a total of 200 time steps immediately following the training sequence. Fig. 3a displays the result of the baseline DSSM when trained solely on the 1st task. Subsequent figures (Figs. 3a–3f) depict predictions for the 1st task after training all 4 tasks using the respective labeled baseline and CLDSSMs. Fig. 3b shows that the observation predictions from the baseline DSSM exhibit a notable disparity from the ground truth, demonstrating its incapability in predicting the trajectory beyond 100 time steps. In contrast, the prediction results of all CLDSSMs consistently align with the ground truth across all 200 time steps, even with a slight mismatch. Notably, the prediction of LwF exhibits superior overall performance with the lowest MSE value after training all tasks, while EWC, MAS, and SI also closely follow the observed rise-and-fall trajectory patterns. The performance difference can be attributed to the different rationales in continual learning methods, as discussed in Section III-C. For example, the EWC method controls parameter deviations, making it adept at capturing trajectory features, while the LwF method utilizes the MSE function to achieve a better holistic result.

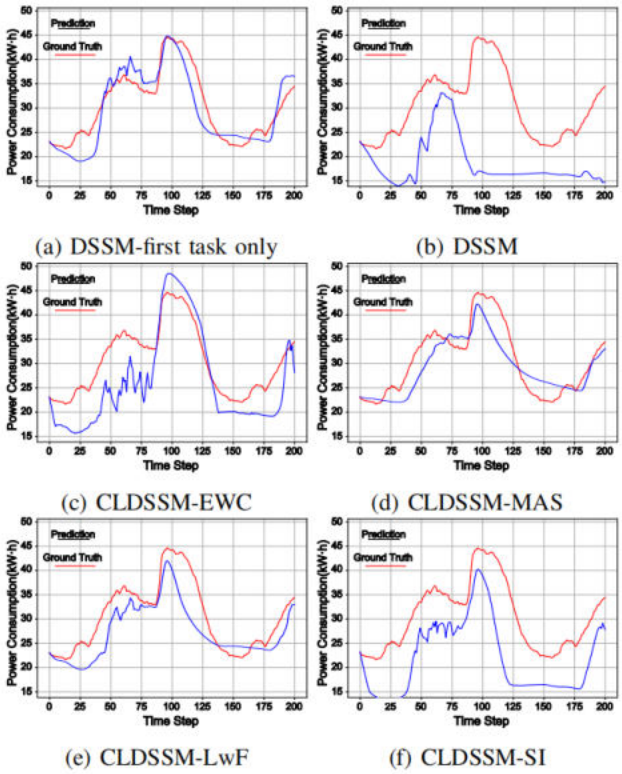
TABLE I: Averaged prediction MSE of the POWER CONSUMPTION dataset. The mean and standard deviation of the prediction results are shown.

	1st Task	2nd Task	3rd Task	4th Task
DSSM	26.15±1.12	56.21±5.35	299.32±19.74	463.36±26.49
EWC	**26.03**±0.98	**25.65**±3.76	58.41±4.84	77.01±7.52
MAS	27.28±1.06	29.26±4.04	55.64±5.18	71.87±6.98
LwF	26.39±1.10	27.19±3.27	**51.77**±4.63	**69.65**±7.26
SI	27.94±1.21	36.44±4.39	64.70±6.59	101.14±9.63

B. Weather Dataset

In this subsection, we conduct another evaluation using a real-world weather dataset comprising 5844 samples with four-dimensional control inputs and one-dimensional observations. This dataset captures the weather information in London from 2005 to 2020, encompassing input features such as cloud cover, sunshine conditions, global radiation, and precipitation. The observed variable is represented by temperature measured in degrees Celsius. Similar to the approach outlined in Section IV-A, we partition the dataset into four segments, each containing 1461 samples spanning four years and serving as an individual task. The sequence length is defined as $T = 50$, and the batch size is set to 22, preserving data for approximately one year as the ground truth for predictions.

Table II presents an overview of different model prediction results for the weather data, evaluated by the MSE between ground truth and predictions across all tasks. Our proposed CLDSSMs exhibit notable effects in mitigating catastrophic forgetting, consistently exceeding the baseline with lower MSE values and more stable standard deviations. Similar to the previous subsection, we also depict in Fig. 4 the prediction results of all the models in the 1st task after four tasks of learning. The 361 test data points come immediately after the training data, with a predicted period spanning approximately



(a) DSSM-first task only (b) DSSM

(c) CLDSSM-EWC (d) CLDSSM-MAS

(e) CLDSSM-LwF (f) CLDSSM-SI

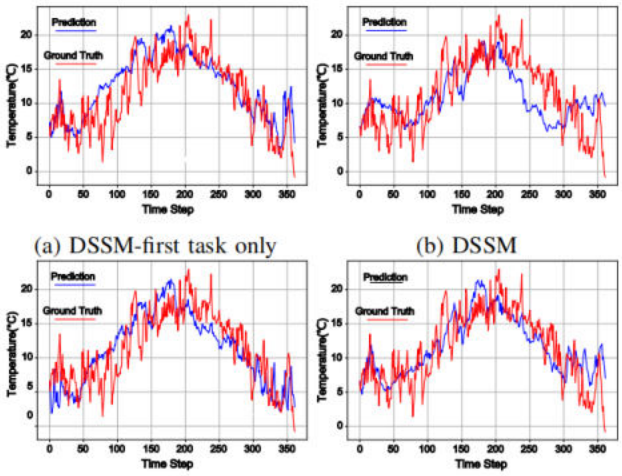
Fig. 3: Prediction results of DSSM and CLDSSMs in the 1st task of the POWER CONSUMPTION dataset.

one year. Notably, the results of CLDSSMs outperform the baseline, with a significant improvement in prediction accuracy at later steps. This is evident in Fig. 4b, where the predictions of the DSSM struggle to converge with the ground truth, eventually failing to follow the trajectory after 200 time steps. In general, similar to the results in the last subsection, the LwF-based CLDSSM demonstrates the lowest prediction MSE after all the training, indicating its ability to achieve the best prediction results.

Lastly, it is noteworthy that our experiments and findings indicate that CLDSSMs demonstrate accelerated convergence when confronted with new tasks during training, surpassing even the predictive performance of baseline DSSM on the current task. This phenomenon is exemplified in Fig. 5, where Fig. 5a and Fig. 5b depict prediction results for the baseline and EWC respectively. Both results represent predictions for the 4th task obtained after training across all 4 tasks. It is evident that the trajectory of CLDSSM-EWC performs globally better and with greater accuracy. Moreover, our empirical results show that the training MSE value for CLDSSM-EWC drops to 18 after 400 epochs of training, while baseline DSSM only reaches a value of around 23 under the same conditions. This notable improvement can be attributed to efficiently handling pre-existing knowledge from previous tasks, facilitating faster adaptation to new ones. In contrast, the absence of regularization in baseline DSSM results in slower training on new tasks and insufficient learning within the same epoch, leading to unsatisfactory test results on all tasks. The ability of such task knowledge extraction in the CLDSSMs highlights the learning efficiency of CLDSSMs from another perspective.

TABLE II: Averaged prediction MSE of the WEATHER dataset. The mean and standard deviation of the prediction results are shown.

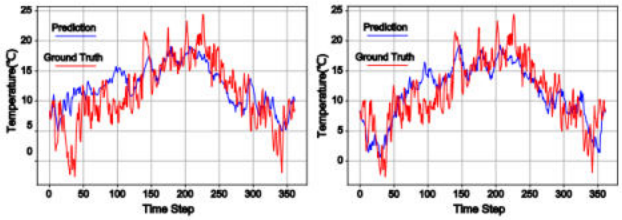
	1st Task	2nd Task	3rd Task	4th Task
DSSM	11.31±0.67	13.12±1.19	16.21±1.83	18.16±1.89
EWC	12.01±0.71	**11.73**±0.95	13.79±1.64	15.34±1.47
MAS	**10.98**±0.59	11.95±1.03	**13.56**±1.36	15.67±1.72
LwF	11.74±0.69	12.29±0.97	13.85±1.29	**14.79**±1.41
SI	11.48±0.54	12.94±1.33	14.62±1.35	16.73±1.64



(a) DSSM-first task only (b) DSSM

(c) CLDSSM-EWC (d) CLDSSM-LwF

Fig. 4: Prediction results of DSSM and CLDSSMs in the 1st task of the WEATHER dataset.



(a) DSSM (b) CLDSSM-EWC

Fig. 5: Prediction results of DSSM and CLDSSM-EWC in the 4th task of the WEATHER dataset.

V. CONCLUSION

This paper introduces a novel class of models called continual learning deep state-space models (CLDSSMs), specifically designed to address the challenge of continual learning in DSSMs. The proposed CLDSSMs demonstrate computational and memory efficiency by incorporating regularization-based methods and leveraging training data from the most recent task. Experiments conducted on various real-world datasets

showcase the effectiveness of CLDSSMs, highlighting their proficiency in multi-task forecasting and superiority in training new tasks. Furthermore, CLDSSMs with EWC and MAS methods exhibit the lowest costs in terms of memory storage and computational complexity, while LwF produces the best forecasting results, achieving the lowest MSE after training on all tasks. These results collectively underscore the excellent performance of CLDSSMs in applications of multiple dynamic systems modeling.

REFERENCES

[1] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge university press, 2013, no. 3.

[2] Y. Zhao, C. Fritsche, G. Hendeby, F. Yin, T. Chen, and F. Gunnarsson, "Cramér–Rao bounds for filtering based on Gaussian process state-space models," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5936–5951, 2019.

[3] A. Xie, F. Yin, B. Ai, S. Zhang, and S. Cui, "Learning while tracking: A practical system based on variational Gaussian process state-space model and smartphone sensory data," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–7.

[4] Y. Zhao, F. Yin, F. Gunnarsson, M. Amirijoo, E. Özkan, and F. Gustafsson, "Particle filtering for positioning based on proximity reports," in *2015 IEEE 18th International Conference on Information Fusion (FUSION)*. IEEE, 2015, pp. 1046–1052.

[5] Y. Zhao, C. Fritsche, F. Yin, F. Gunnarsson, and F. Gustafsson, "Sequential monte carlo methods and theoretical bounds for proximity report based indoor positioning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5372–5386, 2018.

[6] Z. Lin, L. Cheng, F. Yin, L. Xu, and S. Cui, "Output-dependent Gaussian process state-space model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes, Greek, Jun. 2023, pp. 1–5.

[7] Z. Lin, Y. Sun, F. Yin, and A. Thiéry, "Ensemble Kalman filtering meets Gaussian process SSM for non-mean-field and online inference," *arXiv preprint arXiv:2312.05910*, 2023.

[8] Z. Lin, F. Yin, and J. Maroñas, "Towards flexibility and interpretability of Gaussian process state-space model," *arXiv preprint arXiv:2301.08843*, 2023.

[9] Z. Lin, J. Maroñas, Y. Li, F. Yin, and S. Theodoridis, "Towards efficient modeling and inference in multi-dimensional Gaussian process state-space models," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024.

[10] A. M. Alaa and M. van der Schaar, "Attentive state-space modeling of disease progression," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 11 334–11 344.

[11] R. Krishnan, U. Shalit, and D. Sontag, "Structured inference networks for nonlinear state space models," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, California, USA, Feb. 2017, pp. 2101–2109.

[12] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, Quebec, Canada, Dec. 2015, pp. 2980–2988.

[13] A. Klushyn, R. Kurle, M. Soelch, B. Cseke, and P. van der Smagt, "Latent matters: Learning deep state-space models," in *Advances in Neural Information Processing Systems (NeurIPS)*, Virtual, Online, 2021, pp. 10 234–10 245.

[14] D. Gedon, N. Wahlström, T. B. Schön, and L. Ljung, "Deep state space models for nonlinear system identification," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 481–486, 2021.

[15] M. Karl, M. Soelch, J. Bayer, and P. Van der Smagt, "Deep variational Bayes filters: Unsupervised learning of state space models from raw data," in *International Conference on Learning Representations (ICLR)*, Toulon, France, Apr. 2017.

[16] D. Masti and A. Bemporad, "Learning nonlinear state–space models using autoencoders," *Automatica*, vol. 129, p. 109666, 2021.

[17] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 492–503, 2017.

[18] F. Yin, Z. Lin, Q. Kong, Y. Xu, D. Li, S. Theodoridis, and S. Cui, "Fedloc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 187–215, Nov. 2020.

[19] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[20] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2021.

[21] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[22] Z. Chen and B. Liu, *Lifelong machine learning*, 2nd ed. Springer, 2018.

[23] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 348–358.

[24] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2001–2010.

[25] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, Apr. 2018, pp. 3302–3309.

[26] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[27] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *International Conference on Machine Learning (ICML)*, vol. 80, Stockholmsmässan, Stockholm, Sweden, Jul. 2018, pp. 4528–4537.

[28] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[29] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 3987–3995.

[30] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[31] M. K. Titsias, J. Schwarz, A. G. d. G. Matthews, R. Pascanu, and Y. W. Teh, "Functional regularisation for continual learning with Gaussian processes," in *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.

[32] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018.

[33] J. Xu and Z. Zhu, "Reinforced continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Montréal, Canada, Dec. 2018, pp. 907–916.

[34] Y. Chen, D. Sanz-Alonso, and R. Willett, "Autodifferentiable ensemble Kalman filters," *SIAM Journal on Mathematics of Data Science*, vol. 4, no. 2, pp. 801–833, 2022.

[35] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, 2nd ed. Academic Press, 2020.

[36] L. Cheng, F. Yin, S. Theodoridis, S. Chatzis, and T.-H. Chang, "Rethinking Bayesian learning for data analysis: The art of prior and inference in sparsity-aware modeling," *IEEE Signal Processing Magazine*, vol. 39, no. 6, pp. 18–52, Nov. 2022.

[37] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean dynamics*, vol. 53, pp. 343–367, 2003.

[38] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, "The ensemble Kalman filter: a signal processing perspective," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, pp. 1–16, 2017.

[39] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, Apr. 2014.

[40] K. Panousis, S. Chatzis, and S. Theodoridis, "Nonparametric Bayesian deep networks with local competition," in *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, California, USA, Jun. 2019, pp. 4980–4988.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[42] J. J. Rissanen, "Fisher information and stochastic complexity," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 40–47, 1996.

[43] A. Salam and A. E. Hibaoui, "Comparison of machine learning algorithms for the power consumption prediction : - case study of Tetouan city –," in *International Renewable and Sustainable Energy Conference (IRSEC)*, Rabat, Morocco, Dec. 2018, pp. 1–5.